p8106_hw3

Hao Zheng (hz2770)

2022/3/19

```r
# data import
auto_data =
  read.csv("./data/auto.csv") %>%
  mutate(
    origin = as.factor(origin),
    mpg_cat = as.factor(mpg_cat),
    mpg_cat = fct_relevel(mpg_cat, c("low", "high"))
  ) %>%
  na.omit()

set.seed(2022)

indexTrain <- createDataPartition(y = auto_data$mpg_cat, p = 0.7, list = FALSE)
trainData <- auto_data[indexTrain,]
testData <- auto_data[-indexTrain,]
head(trainData)
```

```
##   cylinders displacement horsepower weight acceleration year origin mpg_cat
## 1         8          307        130   3504         12.0   70      1     low
## 2         8          350        165   3693         11.5   70      1     low
## 3         8          318        150   3436         11.0   70      1     low
## 4         8          304        150   3433         12.0   70      1     low
## 5         8          302        140   3449         10.5   70      1     low
## 6         8          429        198   4341         10.0   70      1     low
```

```r
ctrl <- trainControl(method = "repeatedcv", repeats = 5,
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE)
```

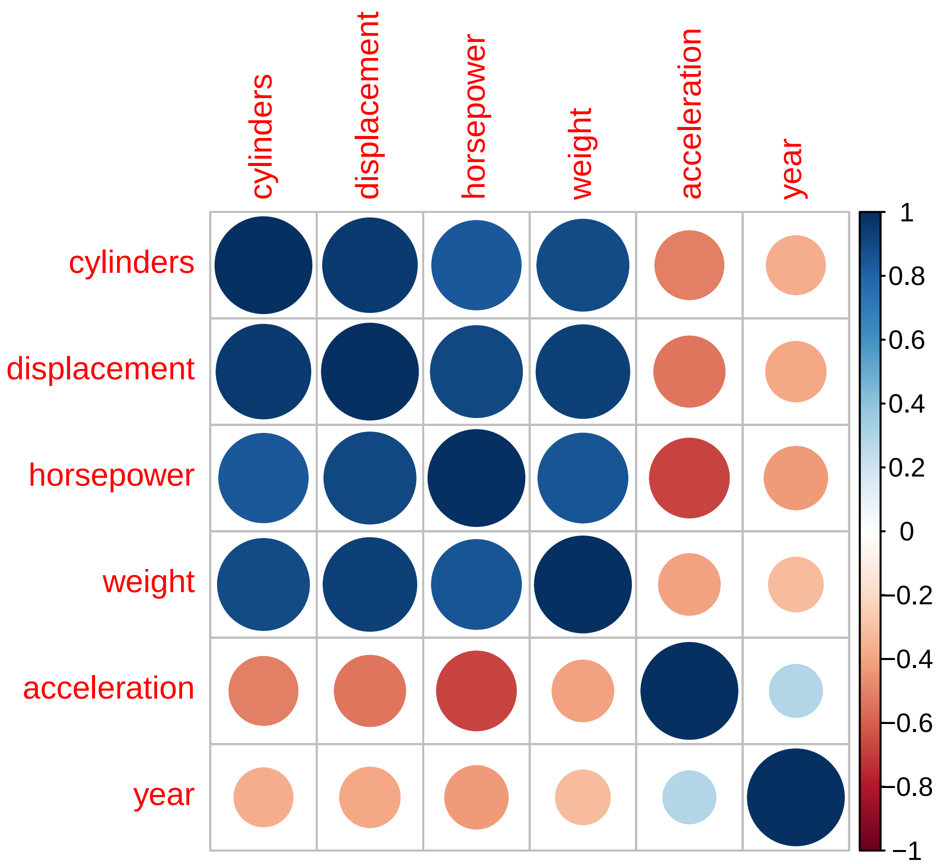
a) Exploratory data analysis

```r
# numeric summary
summary(trainData)
```

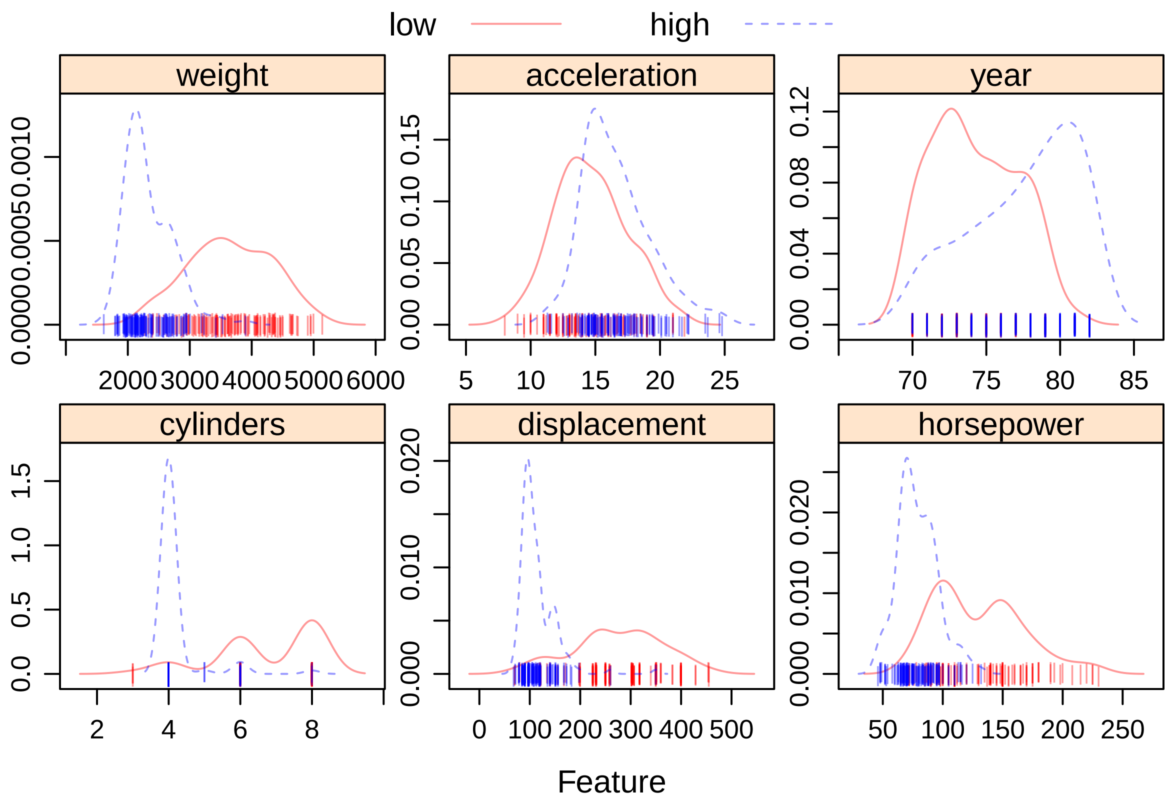
```
##    cylinders      displacement     horsepower        weight      acceleration
##  Min.   :3.00   Min.   : 68.0   Min.   : 46.0   Min.   :1613   Min.   : 8.00
##  1st Qu.:4.00   1st Qu.:100.2   1st Qu.: 75.0   1st Qu.:2222   1st Qu.:13.90
##  Median :4.00   Median :151.0   Median : 95.0   Median :2798   Median :15.50
```

```
##  Mean   :5.46   Mean   :194.3   Mean   :104.5   Mean   :2991   Mean   :15.66
##  3rd Qu.:8.00   3rd Qu.:302.0   3rd Qu.:129.2   3rd Qu.:3635   3rd Qu.:17.32
##  Max.   :8.00   Max.   :455.0   Max.   :230.0   Max.   :5140   Max.   :24.80
##       year       origin  mpg_cat
##  Min.   :70.00   1:175   low :138
##  1st Qu.:73.00   2: 47   high:138
##  Median :76.00   3: 54
##  Mean   :75.92
##  3rd Qu.:79.00
##  Max.   :82.00
```

```r
# correlation Plot
x <- trainData[,1:7]
y <- trainData$mpg_cat
corrplot(cor(x %>% dplyr::select(-origin)), method = "circle", type = "full")
```




```r
# Feature Plot
theme1 <- transparentTheme(trans = .4)
trellis.par.set(theme1)

featurePlot(x %>% dplyr::select(-origin),
            y,
            scales = list(x = list(relation = "free"),
                          y = list(relation = "free")),
            plot = "density", pch ="|",
            auto.key = list(columns = 2))
```



Here, we focus on the training dataset to do explanatory analysis. We have 7 predictors, including 6 numeric variables and 1 factor variable `origin`. The response variable is `mpg_cat`.

From the correlation plot, we can observe that the variables `cylinders`, `displacement`, `horsepower`, `weight` may be positively related with each other, and negatively related to `acceleration`, `year`.

From the feature plot, we see that high MPG may be associated with low weight, large model year, small number of cylinders, small engine displacement and small horsepower.

b) Logistic Regression

```
glm.fit <- glm(mpg_cat ~ .,
               data = auto_data,
               subset = indexTrain,
               family = binomial(link = "logit"))

summary(glm.fit)
```

```
##
## Call:
## glm(formula = mpg_cat ~ ., family = binomial(link = "logit"),
##     data = auto_data, subset = indexTrain)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
```

```
## -2.58449  -0.06036   0.00320   0.16299   2.80615
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -29.776125   8.020420  -3.713 0.000205 ***
## cylinders      0.159497   0.549610   0.290 0.771664
## displacement   0.014037   0.017660   0.795 0.426681
## horsepower    -0.016364   0.029327  -0.558 0.576866
## weight        -0.007198   0.001798  -4.003 6.24e-05 ***
## acceleration   0.114071   0.165285   0.690 0.490103
## year           0.605116   0.120409   5.025 5.02e-07 ***
## origin2        2.285179   0.978723   2.335 0.019551 *
## origin3        1.332239   0.927574   1.436 0.150928
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 382.617  on 275  degrees of freedom
## Residual deviance:  96.539  on 267  degrees of freedom
## AIC: 114.54
##
## Number of Fisher Scoring iterations: 8
```

Fit a glm model using the training data. Among all the predictors, the variables `weight`, `year` and `origin` as European are quite significant.

```
test.pred.prob <- predict(glm.fit, newdata = auto_data[-indexTrain,],
                          type = "response")
test.pred <- rep("low", length(test.pred.prob))
test.pred[test.pred.prob > 0.5] <- "high"
confusionMatrix(data = as.factor(test.pred),
                reference = auto_data$mpg_cat[-indexTrain],
                positive = "high")
```

```
## Warning in confusionMatrix.default(data = as.factor(test.pred), reference =
## auto_data$mpg_cat[-indexTrain], : Levels are not in the same order for reference
## and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction low high
##       low   50    9
##       high   8   49
##
##                Accuracy : 0.8534
##                  95% CI : (0.7758, 0.9122)
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : 1.478e-15
##
##                   Kappa : 0.7069
##
```

```
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8448
##             Specificity : 0.8621
##          Pos Pred Value : 0.8596
##          Neg Pred Value : 0.8475
##              Prevalence : 0.5000
##          Detection Rate : 0.4224
##    Detection Prevalence : 0.4914
##       Balanced Accuracy : 0.8534
##
##        'Positive' Class : high
##
```

From the confusion matrix above, we calculate that correct prediction rate: $(50 + 49)/(50 + 9 + 8 + 49) = 0.8534$.

The confusion matrix also tells us: The no information rate is 0.5, that is the misclassification rate if predict everyone to be positive is 0.5, which is not very ideal. The p-value is 1.478e-15. The sensitivity is 0.8448, specificity is 0.8621. The positive predictive value is 0.8596, negative predictive value is 0.8475.

```r
# logistic model using caret
set.seed(2022)

model.glm = train(x = auto_data[indexTrain, 1:7],
                  y = auto_data$mpg_cat[indexTrain],
                  method = "glm",
                  metric = "ROC",
                  trControl = ctrl)
summary(model.glm)
```

```
##
## Call:
## NULL
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.58449  -0.06036   0.00320   0.16299   2.80615
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -29.776125   8.020420  -3.713 0.000205 ***
## cylinders      0.159497   0.549610   0.290 0.771664
## displacement   0.014037   0.017660   0.795 0.426681
## horsepower    -0.016364   0.029327  -0.558 0.576866
## weight        -0.007198   0.001798  -4.003 6.24e-05 ***
## acceleration   0.114071   0.165285   0.690 0.490103
## year           0.605116   0.120409   5.025 5.02e-07 ***
## origin2        2.285179   0.978723   2.335 0.019551 *
## origin3        1.332239   0.927574   1.436 0.150928
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 382.617  on 275  degrees of freedom
## Residual deviance:  96.539  on 267  degrees of freedom
## AIC: 114.54
##
## Number of Fisher Scoring iterations: 8
```

c) Multivariate adaptive regression spline(MARS)

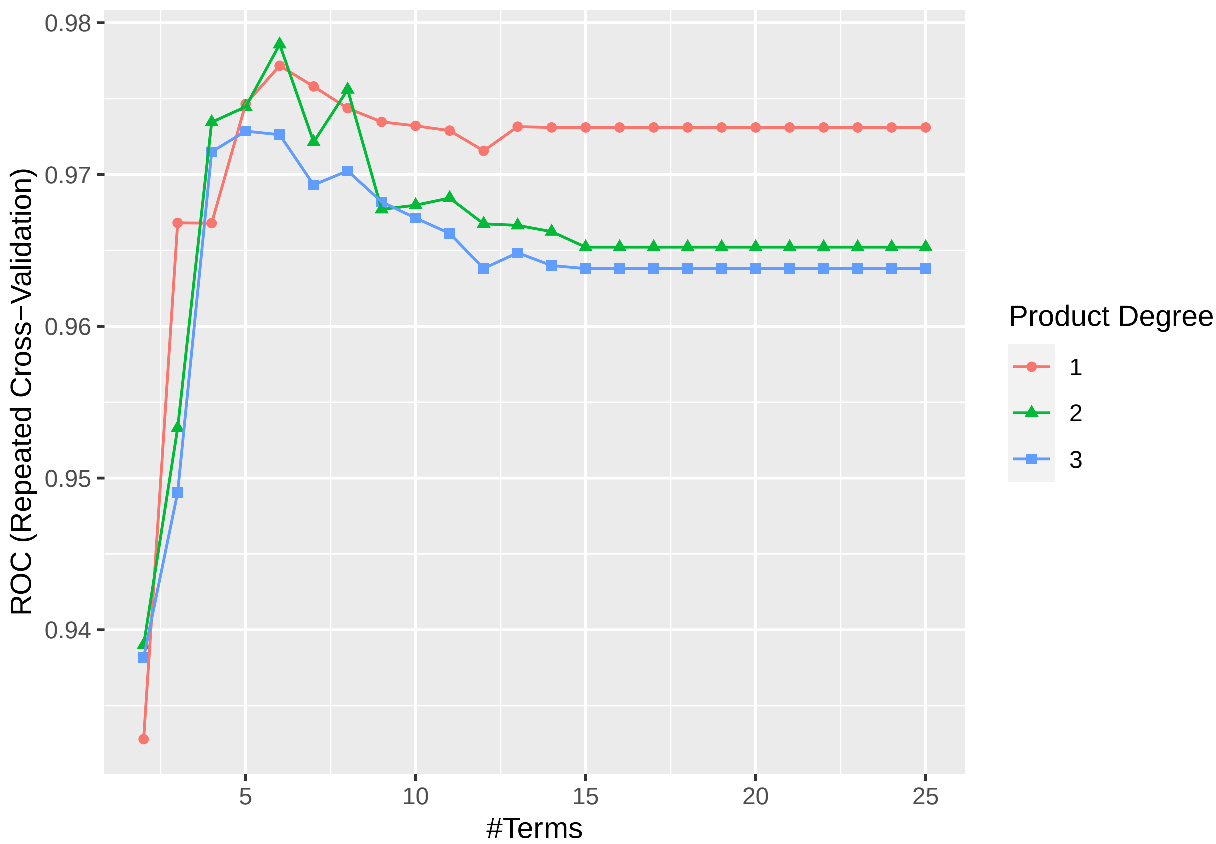
```
mars_grid <- expand.grid(degree = 1:3,
                         nprune = 2:25)
set.seed(2022)
mars.fit <- train(x,
                  y,
                  method = "earth",
                  tuneGrid = mars_grid,
                  trControl = ctrl)
```

```
## Loading required package: earth

## Loading required package: Formula

## Loading required package: plotmo

## Loading required package: plotrix

## Loading required package: TeachingDemos
```

```
summary(mars.fit)
```

```
## Call: earth(x=data.frame[276,7], y=factor.object, keepxy=TRUE,
##             glm=list(family=function.object, maxit=100), degree=2, nprune=6)
##
## GLM coefficients
##                                               high
## (Intercept)                             -7.9803884
## h(250-displacement)                      0.0728756
## h(year-72)                               0.8045225
## h(4-cylinders) * h(250-displacement) -0.1166665
## h(250-displacement) * h(weight-2223) -0.0000415
## h(156-displacement) * h(year-72)     -0.0081274
##
## GLM (family binomial, link logit):
##  nulldev  df       dev  df   devratio     AIC iters converged
##  382.617 275   78.8294 270      0.794   90.83    15         1
##
## Earth selected 6 of 19 terms, and 4 of 8 predictors (nprune=6)
## Termination condition: Reached nk 21
## Importance: displacement, cylinders, year, weight, horsepower-unused, ...
## Number of terms at each degree of interaction: 1 2 3
## Earth GCV 0.06033251    RSS 15.06263    GRSq 0.7604156    RSq 0.781701
```

```r
ggplot(mars.fit)
```




```r
mars.fit$bestTune
```

```
##    nprune degree
## 29      6      2
```

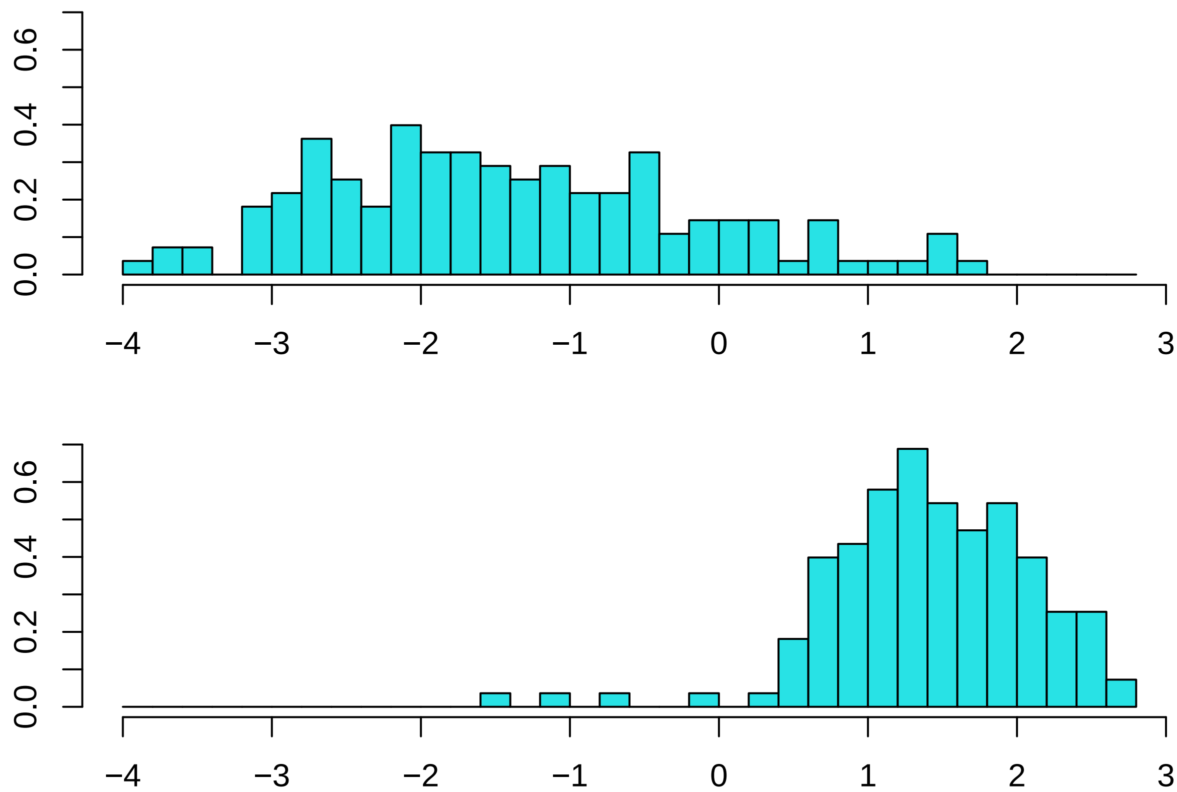
```r
coef(mars.fit$finalModel)
```

```
##                             (Intercept)                h(250-displacement)
##                           -7.980388e+00                       7.287558e-02
##                              h(year-72) h(4-cylinders) * h(250-displacement)
##                            8.045225e-01                      -1.166665e-01
##     h(156-displacement) * h(year-72) h(250-displacement) * h(weight-2223)
##                           -8.127365e-03                      -4.147208e-05
```

Our MARS model select 6 of 19 terms, with 4 out of 8 predictors (nprune = 6). The final model has RSS = 15.06263, R-squared = 0.781701, which is quite big.

d) LDA

```r
set.seed(2022)

lda.fit <- lda(mpg_cat~., data = auto_data,
               subset = indexTrain)

par(mar = rep(2,4))
plot(lda.fit)
```




```r
# The matrix A
lda.fit$scaling
```

```
##                       LD1
## cylinders    -0.234014412
## displacement -0.001889600
## horsepower    0.012887200
## weight       -0.001369713
## acceleration  0.011595656
## year          0.148317904
## origin2       0.571578594
## origin3       0.419477226
```

We perform a LDA fit model. The linear discriminate is plotted above within two classes. Since k=2, we only have k - 1 = 1 linear discriminant.

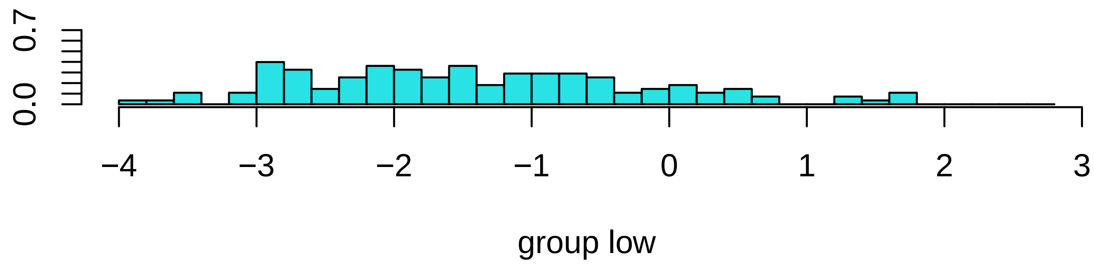
```r
# Use caret to conduct LDA
set.seed(2022)

x = x %>%
  mutate(
    origin = as.numeric(origin)
  )

model.lda <- train(x,
                   y = auto_data$mpg_cat[indexTrain],
                   method = "lda",
                   metric = "ROC",
                   trControl = ctrl)

model.lda$results
```

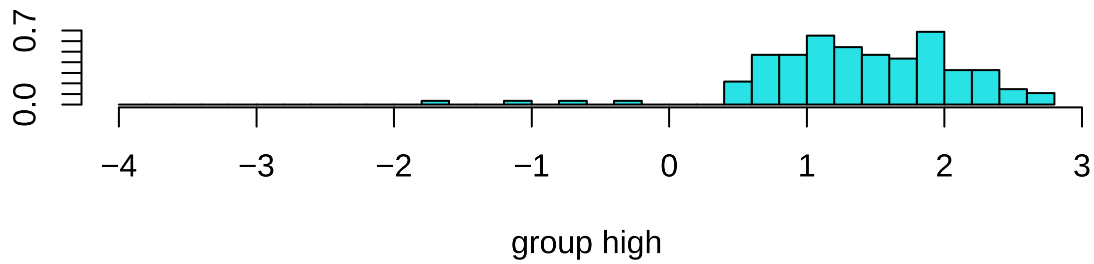
```
##   parameter       ROC      Sens      Spec     ROCSD     SensSD     SpecSD
## 1      none 0.9589566 0.8495604 0.9708791 0.0383611 0.08083545 0.03605979
```

```r
summary(model.lda$finalModel)
```

```
##             Length Class      Mode
## prior        2     -none-     numeric
## counts       2     -none-     numeric
## means       14     -none-     numeric
## scaling      7     -none-     numeric
## lev          2     -none-     character
## svd          1     -none-     numeric
## N            1     -none-     numeric
## call         3     -none-     call
## xNames       7     -none-     character
## problemType  1     -none-     character
## tuneValue    1     data.frame list
## obsLevels    2     -none-     character
## param        0     -none-     list
```

```r
plot(model.lda$finalModel)
```





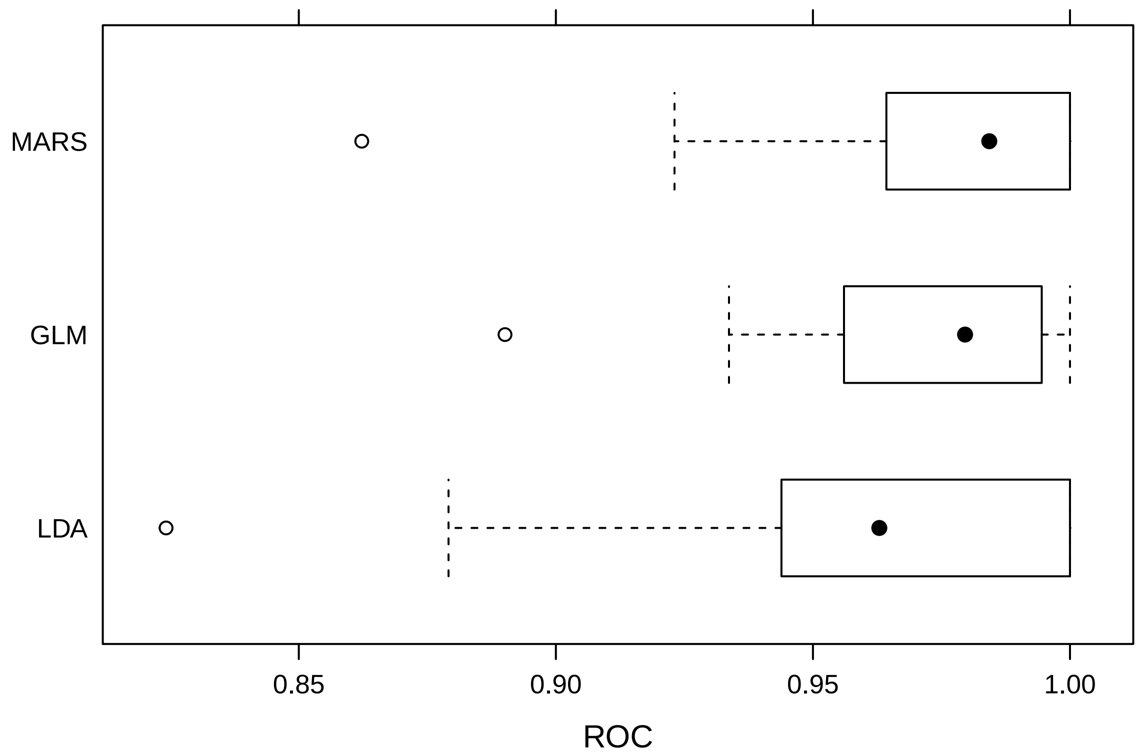
e) Model selection

```r
res <- resamples(list(GLM = model.glm,
                      MARS = mars.fit,
                      LDA = model.lda))
summary(res)
```

```
##
## Call:
## summary.resamples(object = res)
##
## Models: GLM, MARS, LDA
## Number of resamples: 50
##
## ROC
##           Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## GLM  0.8901099 0.9568289 0.9795918 0.9732055 0.9933281    1    0
## MARS 0.8622449 0.9649725 0.9843014 0.9785766 1.0000000    1    0
## LDA  0.8241758 0.9441719 0.9629121 0.9589566 0.9987245    1    0
##
## Sens
##           Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## GLM  0.7142857 0.8571429 0.9285714 0.9018681 0.9285714    1    0
```

```
## MARS 0.7857143 0.9230769 0.9285714 0.9263736 0.9285714    1    0
## LDA  0.7142857 0.7857143 0.8571429 0.8495604 0.9271978    1    0
##
## Spec
##           Min.   1st Qu.    Median      Mean 3rd Qu. Max. NA's
## GLM  0.7692308 0.9230769 0.9285714 0.9372527       1    1    0
## MARS 0.8461538 0.9285714 0.9285714 0.9563736       1    1    0
## LDA  0.9230769 0.9285714 1.0000000 0.9708791       1    1    0
```

```r
bwplot(res, metric = "ROC")
```



Compare the three fit using training data, the MARS model has a rather high ROC.

Now let's plot the ROC curve for MARS model using test data.

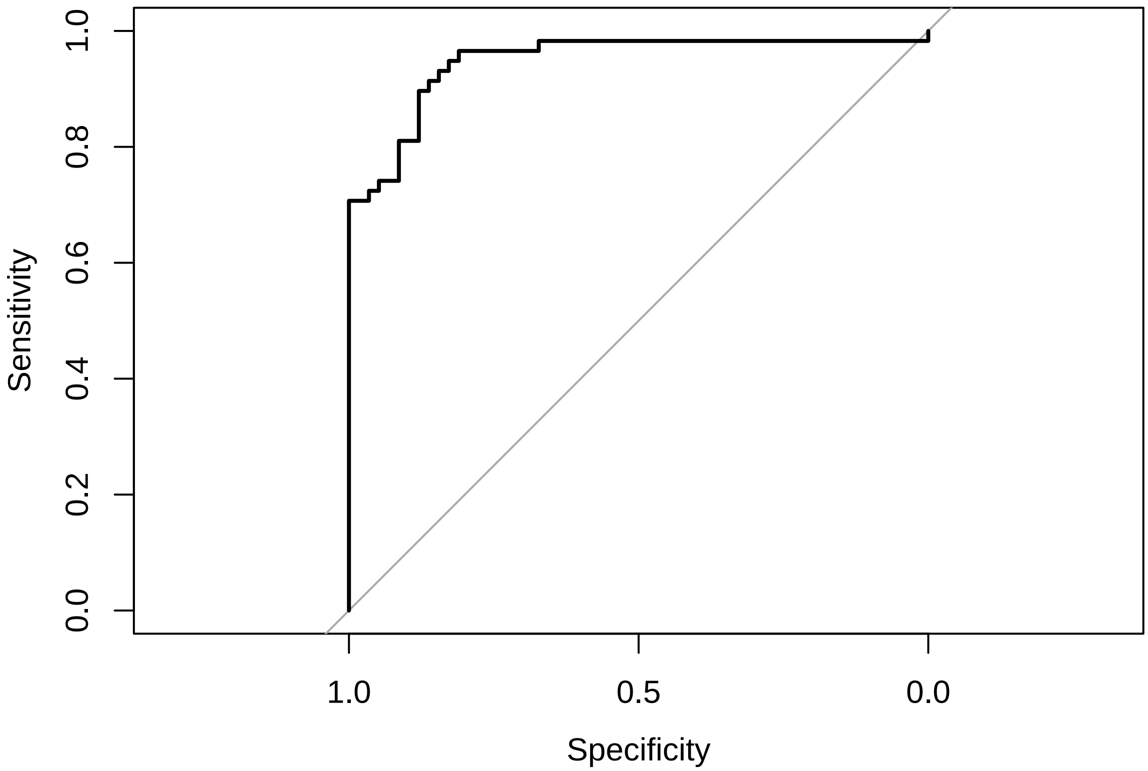
```r
mars.pred <- predict(mars.fit, newdata = auto_data[-indexTrain, 1:7], type = "prob")[,2]
roc.mars <- roc(auto_data$mpg_cat[-indexTrain], mars.pred)
```

```
## Setting levels: control = low, case = high

## Setting direction: controls < cases
```

```r
# AUC
auc_mars <- roc.mars$auc[1];auc_mars
```

```
## [1] 0.9479786
```

```r
plot(roc.mars, legacy.axis = TRUE)
```



The ROC curve of MARS model for the test data is as above. The AUC value is 0.9479786.

```r
test.pred <- rep("low", length(mars.pred))
test.pred[mars.pred > 0.5] <- "high"
confusionMatrix(data = as.factor(test.pred),
                reference = auto_data$mpg_cat[-indexTrain],
                positive = "high")
```

```
## Warning in confusionMatrix.default(data = as.factor(test.pred), reference =
## auto_data$mpg_cat[-indexTrain], : Levels are not in the same order for reference
## and data. Refactoring data to match.
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction low high
##       low   51    7
##       high   7   51
##
##                Accuracy : 0.8793
##                  95% CI : (0.8058, 0.9324)
```

```
##     No Information Rate : 0.5
##     P-Value [Acc > NIR] : <2e-16
##
##                   Kappa : 0.7586
##
##  Mcnemar's Test P-Value : 1
##
##             Sensitivity : 0.8793
##             Specificity : 0.8793
##          Pos Pred Value : 0.8793
##          Neg Pred Value : 0.8793
##              Prevalence : 0.5000
##          Detection Rate : 0.4397
##    Detection Prevalence : 0.5000
##       Balanced Accuracy : 0.8793
##
##        'Positive' Class : high
##
```

The classifications rate of the MARS model on the test data can be calculated by conduct the confusion matrix. The misclassification error rate is 1 - 0.8793 = 0.1207.